UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/691,424	10/22/2003	Michael S. Kaplan	M61.12-0988	6179

27366 7590 11/09/2009
WESTMAN CHAMPLIN (MICROSOFT CORPORATION)
SUITE 1400
900 SECOND AVENUE SOUTH
MINNEAPOLIS, MN 55402

EXAMINER	
COLUCCI, MICHAEL C	
ART UNIT	PAPER NUMBER
2626	
MAIL DATE	DELIVERY MODE
11/09/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/691,424	KAPLAN ET AL.
	Examiner	Art Unit
	MICHAEL C. COLUCCI	2626

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 07/08/2009.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1,2,5-7,10-17,25 and 26 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1,2,5-7,10-17,25 and 26 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments filed 07/08/2009 have been fully considered but they are not persuasive.

Argument 1 (page 11 paragraph 3):

- "Katayama unfortunately uses the word "collating" but it would appear that Katayama does not use that word to mean "ordering" ".

(pages 9 and 13):

- "Claim 1 specifically recites "each compression table pertaining to one of the supported languages and having a compression type identifying a number of symbols in a given compression in the compression table and containing compressions of symbols of that compression type, each compression being a grouping of two or more symbols treated as a single sort element for purposes of linguistic sorting""

Response to argument 1:

Examiner respectfully disagrees. Consider that though Katayama may teach collating appearing to mean matching, regardless, the primary reference of Lisle alone explicitly teaches a collation order and sorting operation assumed for the example of FIG. 2 is that used in the IBM System/370 computer architecture. This is an assigned hierarchical sorting collation order with special characters first in a defined order that is known to users of such systems, followed by the alphabet upper and lower case and

last, by the numerals in the highest collation order of sequence. The collation order may be viewed as equivalent to an overall "alphabetic order" for the possible entries to be sorted. The actual dictionary entries for each dictionary are thus collated first and sorted into the collation order. Each dictionary segment thus begins with some low collation order entry of a given length and a given entry word (or number or character as the case may be) and the segment index ends with the highest collation order entry that appears within that segment of the dictionary being used. The dictionary segment index is used to speed dictionary search time using binary search techniques as will be described (Col. 15 lines 23-63 & Fig. 2)

Further, in view of the amendments to claim 1, newly incorporated reference Edberg also teaches information for the language specific order of collation for a particular language is supplied by language-knowledgeable sources referred to as localizers. The localizers are anyone with language specific information, examples of particular issues associated with various languages are listed in the Background section. In the present invention, the localizers need not provide all the language specific information in a single source as typically required in the current state of the art. The localizers can provide the information in a plurality of sources and the present invention can locate and utilize the information. For example, the localizer with knowledge of the French language can provide information such as collation information for French. This information can be placed in a collation table for access by the present invention. A localizer with knowledge of the Japanese language can provide collation information for

Japanese. This Japanese information can be placed in a separate collation table for access by the present invention. Various language information can be collected in this fashion. (Edberg Col. 7 lines 40-59).

Furthermore, though Lisle uses the term dictionary, Lisle also explicitly teaches *compression tables*, wherein a dictionary and a table are merely structured data in memory. Lisle teaches for example that line A illustrates the start of the header with byte 0 being identified. Bits AA are utilized in the present invention to define the type of single byte compression table that is being used. Two types are possible. Either an existing default value single byte compression table is employed which consists of a prearranged set of assignments for the 256 possible patterns in a single byte table or an alternative existing single byte compression table is to be used and may be found on a predefined external storage medium. Alternatively, a single byte compression table will actually be provided within the compressed text byte string or a dictionary of two or three byte address size could be transmitted. These four possible alternatives are encoded by setting the value of bits AA as follows: if bits AA are 00, a default single byte compression table is used in the remainder of the text compression string which follows the header until a new header appears. If the bit values for AA are 01, a reserved unassigned alternative is employed. This pattern could be used to indicate that the dictionary to be used will be transmitted in the compressed text, Bits 10 for AA define an existing single byte compression table is to be used which should be found by the user on the

predefined external storage medium, such disk or tape. An example would be that a single byte compression table which has been optimized for cancer specialists in the medical field, for example, is to be employed for a given text compression. The final alternative where bits AA are 11, is used to indicate that a single byte compression table has actually been generated and will be supplied within the compression string which follows. The remainder of the bits B through G in byte 0, line A, FIG. 6, are utilized as indicators to define which types or choices of other word dictionaries are employed in the compression routine for the given stream of compressed text that follows the header (Lisle Col. 7 lines 32-67).

Additionally, Okada improves the teachings of Lisle in view of Katayama, wherein Okada teaches discriminating the kind of language, wherein a separating unit 32 per language is provided for the language string separating unit 12. On the basis of the discrimination result of the language by the row octet decoder 30, the separating unit 32 per language separates the input Unicode data into each language string such as Latin (English), Greek, or the like. A compressing unit corresponding to each language allocated for the Unicode is individually provided for the language string compressing unit 14. In the embodiment, a Latin compressing unit 34, a Greek compressing unit 36, a Hangul compressing unit 38, a Kanji compressing unit 40, and the like are provided. As a compressing unit per language which is provided for the language string compressing unit 14, it is sufficient to properly decide the compressing unit in accordance with the language which is treated in the Unicode source data as a

compression target. The compression data per language compressed by each of the Latin compressing unit 34, Greek compressing unit 36, Hangul compressing unit 38, and Kanji compressing unit 40 is unified by a code unifying unit 42 and the unified data is outputted as compression data. As a compressing method of each compressing unit per language provided for the language string compressing unit 14, a plurality of dictionary memories corresponding to the languages are provided and there is executed a Ziv-Lempel encoding for encoding by a longest coincidence retrieval of the character string which is inputted per data of the language string and the character string which has already been registered in the dictionary for every language. In the Ziv-Lempel encoding, any one of the dynamic dictionary method and the slide dictionary method can be used. As another compressing method, for the character string separated every language, on the basis of a probability table per language string obtained until now, the character string which is inputted every data can be also multi-value arithmetic encoded. The source data of the Unicode in which different languages mixed exist is separated every language and is individually compressed, so that the compression of each character string in which statistic natures are similar is executed. A compressing function in the Ziv-Lempel encoding, arithmetic encoding, or the like is effectively used and a high compression ratio can be realized (Okada Col. 12 line 26 –Col. 13 line 8 & Fig. 17 and 18 compression, decompression).

Consider that Lisle also teaches the identification of characters to form words for the purposes of text compression, wherein a text compression method and apparatus

are disclosed that enable overall compression ratios of more than six or eight to one for normal language text. Plural multiple-word dictionaries that are specialized for the particular field of use are employed together with a header transmission format that identifies which dictionaries are to be used. In addition, entries in these dictionaries are categorized by a weighted frequency of use ranking in which the product of the word length in characters and the frequency of occurrence of that word in the text is taken as the weighted figure of merit for ranking words to be placed in the individual dictionaries (Abstract).

Further, for the above reasons Examiner has maintained the use of Lisle in view of Katayama, Okada, and Edberg to address claims 25 and 26 as part of the rejection for claim 11.

Additionally, Examiner takes Official Notice that it is well known to compress input characters/symbols into a different representation such as the dictionary methods (LZX, LZ78, LZFG, LZRW1, LZRW4, LZW, LZMW, LZAP, LZY,LZ, and other variants of dictionary compression methods taught by "*David Solomon, Data Compression – The Complete Reference, 4th Ed.*, Pages 172-224 & Tables 3.25 and 3.26" demonstrate input and output variants,- wherein single symbols/characters are sorted and represented as a combination of symbols/characters."

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claim 1, 2, 5-7, 10-12, 15-17, 25, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lisle et al US 4,843,389 (hereinafter Lisle) in view of Katayama et al. US 6260051 B1 US 5550541 A (hereinafter Katayama) and further in view of Okada US 5889481 A (hereinafter Okada) and Edberg 5,873,111 A (hereinafter Edberg).

Re claims 1 and 6, Lisle teaches a computer-readable medium having computer-executable instructions for performing a method for building a symbol table for storing sort weights for a plurality of linguistic symbols used in a plurality of languages supported by a computer system (Col. 15 lines 45-63), the method comprising:

constructing the symbol table (Col. 19 lines 36-59) to contain a list of code points (Col. 20 lines 35-56) each uniquely identifying one of the symbols, and a sort weight for the symbol identified by said each code point (Col. 15 lines 45-63);

providing a plurality of compression tables, each compression table pertaining to one of the supported languages and having a compression type and containing compressions of symbols of that compression type

for each code point in the symbol table (Col. 20 lines 35-56), sorting the compression tables using a processor of the computer,(Col. 19 lines 36-59) to identify a highest compression type our compressions beginning with the symbol (Col. 15 lines 45-63) identified by said each code point (Col. 20 lines 35-56);

storing in the symbol table a tag for the code point to indicate said highest compression type for the code point (Col. 20 lines 35-56).

wherein the tag for each code point is stored as a portion of the sort weight of the symbol identified by said each code point, and wherein the sort weight of the symbol identified by said each code point comprises a case weight value (Col. 15 lines 45-63), and wherein the tag for said each code point is stored as part of the case weight value for said each code point (Col. 20 lines 35-56)

NOTE: Tagging a code point is construed to be both functionally equivalent and equally effective as ranking or ordering a code point or address in memory for the purposes of a hierarchical classification.

However, Lisle fails to teach each compression being a grouping of two or more symbols treated as a single unit for purposes of linguistic sorting and the compression type identifying a number of symbols in a given compression in the compression table

Katayama teaches a registration two-character chain table producing unit 194 for producing a first table block, in which a plurality of registration first and second two-character chains respectively including the same type of for general character and the position numbers of the registration first and second two-character chains are listed in

the order of arranging the chains in the converted registration character string, for each fore general character type, producing a second table block, in which a plurality of registration special two-character chains respectively including the same type of fore symbolic character and the position numbers of the registration special two-character chains are listed in the order of arranging the chains in the converted registration character string, for each fore symbolic character type, and combining each first table block corresponding to one type of fore general character and one second table block corresponding to one type of fore symbolic character determined in correspondence to the type of the fore general character to form a two-character chain table for each character group, the fore characters of the chains in each two-character chain table belonging to the same character group (Katayama Col. 130 lines 33-53).

Further, Katayama teaches a character chain collating and judging unit 200 for receiving the position numbers of one particular two-character chain Tc1 from the storing unit 195 just after the reception of the position numbers of another particular two-character chain Tc2 under the control of the control unit 199. (First collation case), collating each position number of a particular second two-character chain Tc1 with a particular position number of a particular first two-character chain Tc2 to judge whether or not each position number of the particular second two-character chain Tc1 agrees with the particular position number of the particular first two-character chain Tc2 (second collation case), collating each position number of a particular special two-character chain Tc1 with a particular position number of a particular first two-character chain Tc2 to judge whether or not each position number of the particular special two-

character chain Tc1 is higher than the particular position number of the particular first two-character chain Tc2 by one (third collation case), collating each position number of a particular special two-character chain Tc1 with a particular position number of a particular second two-character chain Tc2 to judge whether or not each position number of the particular special two-character chain Tc1 is higher than the particular position number of the particular second two-character chain Tc2 by two (fourth collation case), collating each position number of a particular first two-character chain Tc1 with a particular position number of a particular special two-character chain Tc2 to judge whether or not each position number of the particular first two-character chain Tc1 is higher than the particular position number of the particular special two-character chain Tc2 by one (fifth collation case), and detecting a particular position number of a particular two-character chain of the particular two-character chain table Tc1 for each collation case (Katayama Col. 131 lines 27-67).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Lisle to incorporate compression being a grouping of two or more symbols treated as a single unit for purposes of linguistic sorting and the compression type identifying a number of symbols in a given compression as taught by Katayama to allow for control of several letters/symbols within a character chain, wherein a table is used to track, judge, and find the position and number of symbols present (Katayama Col. 130 lines 33-53).

However, Lisle in view of Katayama fails to teach

providing a plurality of compression tables, each compression table pertaining to *one of the supported languages* and having a compression type and containing compressions of symbols of that compression type

linguistic sorting determined based on a compression type of the given compression, a first of the two or more symbols in the given compression and a predefined order of symbols

Okada teaches discriminating the kind of language, wherein a separating unit 32 per language is provided for the language string separating unit 12. On the basis of the discrimination result of the language by the row octet decoder 30, the separating unit 32 per language separates the input Unicode data into each language string such as Latin (English), Greek, or the like. A compressing unit corresponding to each language allocated for the Unicode is individually provided for the language string compressing unit 14. In the embodiment, a Latin compressing unit 34, a Greek compressing unit 36, a Hangul compressing unit 38, a Kanji compressing unit 40, and the like are provided. As a compressing unit per language which is provided for the language string compressing unit 14, it is sufficient to properly decide the compressing unit in accordance with the language which is treated in the Unicode source data as a compression target. The compression data per language compressed by each of the Latin compressing unit 34, Greek compressing unit 36, Hangul compressing unit 38, and Kanji compressing unit 40 is unified by a code unifying unit 42 and the unified data is outputted as compression data. As a compressing method of each compressing unit per language provided for the language string compressing unit 14, a plurality of

dictionary memories corresponding to the languages are provided and there is executed a Ziv-Lempel encoding for encoding by a longest coincidence retrieval of the character string which is inputted per data of the language string and the character string which has already been registered in the dictionary for every language. In the Ziv-Lempel encoding, any one of the dynamic dictionary method and the slide dictionary method can be used. As another compressing method, for the character string separated every language, on the basis of a probability table per language string obtained until now, the character string which is inputted every data can be also multi-value arithmetic encoded. The source data of the Unicode in which different languages mixed exist is separated every language and is individually compressed, so that the compression of each character string in which statistic natures are similar is executed. A compressing unction in the Ziv-Lempel encoding, arithmetic encoding, or the like is effectively used and a high compression ratio can be realized (Okada Col. 12 line 26 –Col. 13 line 8 & Fig. 17 and 18 compression, decompression).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Lisle in view of Katayama to incorporate providing a plurality of compression tables, each compression table pertaining to one of the supported languages and having a compression type and containing compressions of symbols of that compression type and linguistic sorting determined based on a compression type of the given compression, a first of the two or more symbols in the given compression and a predefined order of symbols as taught by Okada to allow for the distinguishing between multiple languages mixed or separately in a Unicode

environment dealing with compression, decompression, and sorting to create a high compression ratio/type depending on the language (i.e. grammar), wherein compression ratios vary for each language in reference to a already existing language (Okada Col. 12 line 26 –Col. 13 line 8 & Fig. 17 and 18 compression, decompression).

However, Lisle, in view of Katayama and Okada fails to teach a tag associated with a code point

Edberg teaches character attributes that may be organized in a particular collation order such that information located earlier in the list indicate a higher priority level of significance. For example, if "number" comes before "letter" in the order of the character attributes in class 40, then any number will be collated before any letter, such that "10" will be listed before "apple" in a list of information which has been collated by the sample ordering of category 32a. Alternatively, the character attributes 46 may be tagged with a prefix 43. The lower the prefix 43 of a character attribute 46, the earlier it places in the collation order. For example, in the Unicode category 32c, Latin letters would list before Cyrillic letters in a collation order (Edberg Col. 12 lines 7-12).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Lisle in view of Katayama and Okada to incorporate a tag for each code point is stored as a portion of the sort weight of the symbol identified by said each code point, and wherein the sort weight of the symbol identified by said each code point comprises a case weight value and wherein the tag for said each code point is stored as part of the case weight value for said each code

point as taught by Edberg to allow for proper ordering and collation of characters, wherein prefixes are considered in a language specific text (i.e. Unicode and/or Latin), and are tagged with a grammatical element such as prefix as part of a collation order (Edberg Col. 12 lines 7-12).

Re claims 11, 16, 25, and 26 Lisle teaches a computer-readable medium having computer-executable instructions for performing steps for a computer search program to carry out a linguistic sorting operation (Col. 15 lines 45-63, comprising:

receiving an input string containing a plurality linguistic symbols (Col. 6 lines 42-58) used in a given language (Col. 15 lines 45-63);

for a first symbol in a combination of symbols in the input string (Col. 15 lines 45-63), referencing a symbol table (Col. 20 lines 35-56) to obtain a highest compression type for compressions beginning with said first symbol, the symbol table having a list of code points each uniquely identifying a symbol and a sort weight for the symbol identified by said each code point;

performing a binary search (Col. 16 lines 6-27) through each of a plurality of compression tables (Col. 19 lines 36-59) containing compressions for the given language to find a matching compression that matches said combination of symbols in the input string (Col. 16 lines 6-27), wherein the plurality of compression tables are searched in a descending order (Col. 15 lines 45-63) of compression types of the compression tables (Col. 19 lines 36-59) starting with a compression table having a

compression type equal to said highest compression type for said first symbol (Col. 15 lines 45-63).

NOTE: Tagging a code point is construed to be both functionally equivalent and equally effective as ranking or ordering a code point or address in memory for the purposes of a hierarchical classification.

NOTE: *Lisle alone explicitly teaches a collation order assumed for the example of FIG. 2 is that used in the IBM System/370 computer architecture. This is an <u>assigned hierarchical sorting collation order</u> with special characters first in a defined order that is known to users of such systems, followed by the alphabet upper and lower case and last, by the numerals in the <u>highest collation order of sequence</u>. The collation order may be viewed as equivalent to an overall "alphabetic order" for the possible entries to be sorted. The actual dictionary entries for each dictionary are thus collated first and sorted into the collation order. Each dictionary segment thus begins with some low collation order entry of a given length and a given entry word (or number or character as the case may be) and the segment index ends with the <u>highest collation order entry that appears within that segment of the dictionary being used</u>. The dictionary <u>segment index is used to speed dictionary search time</u> using binary search techniques as will be described (Col. 15 lines 23-63 & Fig. 2).*

However, Lisle fails to teach each compression being a grouping of two or more symbols treated as a single unit for purposes of linguistic sorting and the compression type identifying a number of symbols in a given compression

Katayama teaches a registration two-character chain table producing unit 194 for producing a first table block, in which a plurality of registration first and second two-character chains respectively including the same type of for general character and the position numbers of the registration first and second two-character chains are listed in the order of arranging the chains in the converted registration character string, for each fore general character type, producing a second table block, in which a plurality of registration special two-character chains respectively including the same type of fore symbolic character and the position numbers of the registration special two-character chains are listed in the order of arranging the chains in the converted registration character string, for each fore symbolic character type, and combining each first table block corresponding to one type of fore general character and one second table block corresponding to one type of fore symbolic character determined in correspondence to the type of the fore general character to form a two-character chain table for each character group, the fore characters of the chains in each two-character chain table belonging to the same character group (Katayama Col. 130 lines 33-53).

Further, Katayama teaches a character chain collating and judging unit 200 for receiving the position numbers of one particular two-character chain Tc1 from the storing unit 195 just after the reception of the position numbers of another particular two-character chain Tc2 under the control of the control unit 199. (First collation case), collating each position number of a particular second two-character chain Tc1 with a particular position number of a particular first two-character chain Tc2 to judge whether or not each position number of the particular second two-character chain Tc1 agrees

with the particular position number of the particular first two-character chain Tc2 (second collation case), collating each position number of a particular special two-character chain Tc1 with a particular position number of a particular first two-character chain Tc2 to judge whether or not each position number of the particular special two-character chain Tc1 is higher than the particular position number of the particular first two-character chain Tc2 by one (third collation case), collating each position number of a particular special two-character chain Tc1 with a particular position number of a particular second two-character chain Tc2 to judge whether or not each position number of the particular special two-character chain Tc1 is higher than the particular position number of the particular second two-character chain Tc2 by two (fourth collation case), collating each position number of a particular first two-character chain Tc1 with a particular position number of a particular special two-character chain Tc2 to judge whether or not each position number of the particular first two-character chain Tc1 is higher than the particular position number of the particular special two-character chain Tc2 by one (fifth collation case), and detecting a particular position number of a particular two-character chain of the particular two-character chain table Tc1 for each collation case (Katayama Col. 131 lines 27-67).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Lisle to incorporate compression being a grouping of two or more symbols treated as a single unit for purposes of linguistic sorting and the compression type identifying a number of symbols in a given compression as taught by Katayama to allow for control of several letters/symbols within

a character chain, wherein a table is used to track, judge, and find the position and number of symbols present (Katayama Col. 130 lines 33-53).

However, Lisle in view of Katayama fails to teach a highest compression type for compressions beginning with said first symbol, the symbol table having a list of code points each uniquely identifying a symbol and a sort weight for the symbol identified by said each code point for a given language.

Okada teaches discriminating the kind of language, wherein a separating unit 32 per language is provided for the language string separating unit 12. On the basis of the discrimination result of the language by the row octet decoder 30, the separating unit 32 per language separates the input Unicode data into each language string such as Latin (English), Greek, or the like. A compressing unit corresponding to each language allocated for the Unicode is individually provided for the language string compressing unit 14. In the embodiment, a Latin compressing unit 34, a Greek compressing unit 36, a Hangul compressing unit 38, a Kanji compressing unit 40, and the like are provided. As a compressing unit per language which is provided for the language string compressing unit 14, it is sufficient to properly decide the compressing unit in accordance with the language which is treated in the Unicode source data as a compression target. The compression data per language compressed by each of the Latin compressing unit 34, Greek compressing unit 36, Hangul compressing unit 38, and Kanji compressing unit 40 is unified by a code unifying unit 42 and the unified data is outputted as compression data. As a compressing method of each compressing unit

per language provided for the language string compressing unit 14, a plurality of dictionary memories corresponding to the languages are provided and there is executed a Ziv-Lempel encoding for encoding by a longest coincidence retrieval of the character string which is inputted per data of the language string and the character string which has already been registered in the dictionary for every language. In the Ziv-Lempel encoding, any one of the dynamic dictionary method and the slide dictionary method can be used. As another compressing method, for the character string separated every language, on the basis of a probability table per language string obtained until now, the character string which is inputted every data can be also multi-value arithmetic encoded. The source data of the Unicode in which different languages mixed exist is separated every language and is individually compressed, so that the compression of each character string in which statistic natures are similar is executed. A compressing unction in the Ziv-Lempel encoding, arithmetic encoding, or the like is effectively used and a high compression ratio can be realized (Okada Col. 12 line 26 –Col. 13 line 8 & Fig. 17 and 18 compression, decompression).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Lisle in view of Katayama to incorporate a highest compression type for compressions beginning with said first symbol, the symbol table having a list of code points each uniquely identifying a symbol and a sort weight for the symbol identified by said each code point for a given language as taught by Okada to allow for the distinguishing between multiple languages mixed or separately in a Unicode environment dealing with compression, decompression, and sorting to create

a high compression ratio/type depending on the language (i.e. grammar), wherein compression ratios vary for each language in reference to a already existing language (Okada Col. 12 line 26 –Col. 13 line 8 & Fig. 17 and 18 compression, decompression).

Re claim 12, Lisle teaches a computer-readable medium as in claim 11, wherein the compressions in each of the compression tables (Col. 19 lines 36-59) are sorted according to code points for symbols forming the compressions (Col. 15 lines 45-63).

Re claim 2, 7, and 15, Lisle in view of Katayama fails to teach the computer-readable medium as in claim 1, wherein the code points are assigned to the symbols according to the Unicode standard.

Okada teaches discriminating the kind of language, wherein a separating unit 32 per language is provided for the language string separating unit 12. On the basis of the discrimination result of the language by the row octet decoder 30, the separating unit 32 per language separates the input Unicode data into each language string such as Latin (English), Greek, or the like. A compressing unit corresponding to each language allocated for the Unicode is individually provided for the language string compressing unit 14. In the embodiment, a Latin compressing unit 34, a Greek compressing unit 36, a Hangul compressing unit 38, a Kanji compressing unit 40, and the like are provided. As a compressing unit per language which is provided for the language string compressing unit 14, it is sufficient to properly decide the compressing unit in accordance with the language which is treated in the Unicode source data as a

compression target. The compression data per language compressed by each of the Latin compressing unit 34, Greek compressing unit 36, Hangul compressing unit 38, and Kanji compressing unit 40 is unified by a code unifying unit 42 and the unified data is outputted as compression data. As a compressing method of each compressing unit per language provided for the language string compressing unit 14, a plurality of dictionary memories corresponding to the languages are provided and there is executed a Ziv-Lempel encoding for encoding by a longest coincidence retrieval of the character string which is inputted per data of the language string and the character string which has already been registered in the dictionary for every language. In the Ziv-Lempel encoding, any one of the dynamic dictionary method and the slide dictionary method can be used. As another compressing method, for the character string separated every language, on the basis of a probability table per language string obtained until now, the character string which is inputted every data can be also multi-value arithmetic encoded. The source data of the Unicode in which different languages mixed exist is separated every language and is individually compressed, so that the compression of each character string in which statistic natures are similar is executed. A compressing unction in the Ziv-Lempel encoding, arithmetic encoding, or the like is effectively used and a high compression ratio can be realized (Okada Col. 12 line 26 –Col. 13 line 8 & Fig. 17 and 18 compression, decompression).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Lisle in view of Katayama to incorporate a Unicode standard for assigning code points to symbols as taught by Okada to allow for

the distinguishing between multiple languages mixed or separately in a Unicode environment dealing with compression, decompression, and sorting to create a high compression ratio/type depending on the language (i.e. grammar), wherein compression ratios vary for each language in reference to a already existing language (Okada Col. 12 line 26 –Col. 13 line 8 & Fig. 17 and 18 compression, decompression).

Re claim 17, Lisle teaches the computer-readable medium as in claim 11, having further computer-executable instructions for storing a sort weight (Col. 15 lines 45-63) for the matching compression (Col. 16 lines 6-27).

Re claims 5 and 10, Lisle teaches the computer-readable medium as in claim 1, further comprising computer-executable instructions for performing steps of sorting compressions (Col. 15 lines 45-63) in each of the compression tables based on combinations of code points (Col. 20 lines 35-56) of the compressions in said each compression table (Col. 19 lines 36-59).

Re claim 13, Lisle teaches computer-readable medium as in claim 12, wherein each code point in the symbol table includes a tag indicating a highest compression type (Col. 19 lines 36-59) for said each code point (Col. 20 lines 35-56), and wherein said step of referencing retrieves the tag for the code point identifying said first symbol (Col. 15 lines 45-63).

Re claim 14, Lisle teaches sort weight of the symbol (Col. 15 lines 45-63) identified by said each code point (Col. 20 lines 35-56).

However Lisle in view of Katayama and Okada fails to teach the computer-readable medium as in claim 1, wherein the tag for each code point is stored as a portion

Edberg teaches character attributes that may be organized in a particular collation order such that information located earlier in the list indicate a higher priority level of significance. For example, if "number" comes before "letter" in the order of the character attributes in class 40, then any number will be collated before any letter, such that "10" will be listed before "apple" in a list of information which has been collated by the sample ordering of category 32a. Alternatively, the character attributes 46 may be tagged with a prefix 43. The lower the prefix 43 of a character attribute 46, the earlier it places in the collation order. For example, in the Unicode category 32c, Latin letters would list before Cyrillic letters in a collation order (Edberg Col. 12 lines 7-12).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Lisle in view of Katayama and Okada to incorporate the tag for each code point stored as a portion as taught by Edberg to allow for proper ordering and collation of characters, wherein prefixes are considered in a language specific text (i.e. Unicode and/or Latin) (Edberg Col. 12 lines 7-12).

Conclusion

4. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael C. Colucci whose telephone number is (571)-270-1847. The examiner can normally be reached on 9:30 am - 6:00 pm, Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Richemond Dorvil can be reached on (571)-272-7602. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Michael C Colucci/
Examiner, Art Unit 2626
Patent Examiner
AU 2626
(571)-270-1847
Examiner FAX: (571)-270-2847
Michael.Colucci@uspto.gov

/Richemond Dorvil/
Supervisory Patent Examiner, Art Unit 2626